

FASTER DISCRETE COSINE TRANSFORMS USING SCALED TERMS

CROSS-REFERENCE TO RELATED APPLICATION

This application is related to the following co-pending and commonly-
5 assigned patent applications, which are hereby incorporated herein by reference in
their respective entirety:

"FASTER TRANSFORMS USING SCALED TERMS" to Trelewicz et.al.,
having attorney docket no. BLD9-2000-0059US1.

"FASTER TRANSFORMS USING EARLY ABORTS AND PRECISION
10 REFINEMENTS" to Mitchell et al., having attorney docket no. BLD9-2000-0064US1.

BACKGROUND OF THE INVENTION

15 1. Field of the Invention.

This invention relates in general to data processing, and more particularly to
faster discrete cosine transforms using scaled terms.

2. Description of Related Art.

20 Transforms, which take data from one domain (e.g., sampled data) to another
(e.g., frequency space), are used in many signal and/or image processing
applications. Such transforms are used for a variety of applications, including, but
not limited to data analysis, feature identification and/or extraction, signal correlation,

data compression, or data embedding. Many of these transforms require efficient implementation for real-time and/or fast execution.

Data compression is desirable in many data handling processes, where too much data is present for practical applications using the data. Commonly,

5 compression is used in communication links, to reduce transmission time or required bandwidth. Similarly, compression is preferred in data storage systems, including digital printers and copiers, where "pages" of a document to be printed may be stored temporarily in memory. Here the amount of media space can be substantially reduced with compression. Generally speaking, scanned images, i.e., electronic
10 representations of hard copy documents, are often large, and thus make desirable candidates for compression.

It is well-known in the art to use a discrete cosine transform for data compression. In particular examples referred to herein, the terms images and image processing will be used. However, those skilled in the art will recognize that the
15 present invention is not meant to be limited to processing images but is applicable to processing different data, such as audio data, scientific data, image data, etc.

In combination with other techniques, such as color subsampling, quantization, Huffman coding and run-length coding, the discrete cosine transform can compress a digital color image by a factor of approximately thirty to one with
20 virtually no noticeable image degradation. Because of its usefulness in data compression, the discrete cosine transform is an integral part of several data compression standards used by international standards committees such as the International Standards Organization.

DCT (Discrete Cosine Transform), disseminated by the JPEG committee, is a lossy compression system which reduces data redundancies based on pixel to pixel correlations. Generally, an image does not change very much on a pixel to pixel basis and therefore has what is known as "natural spatial correlation". In natural scenes, correlation is generalized, but not exact. Noise makes each pixel somewhat different from its neighbors. Moreover, signal and data processing frequently needs to convert the input data into transform coefficients for the purposes of analysis. Often only a quantized version of the coefficients are needed (e.g., JPEG/MPEG data compression or audio/voice compression). Many such applications need to be done fast in real time such as the generation of JPEG data for high speed printers.

Generally, an example of a JPEG DCT compression and decompression system may be had by referencing the Encyclopedia of Graphics File Formats, by J. D. Murray and W. vanRyper, pp. 159-171 (1994, O'Reilly & Associates, Inc.). Further description of the draft JPEG standard may be found, for example, in "JPEG Still Image Data Compression Standard," by W. Pennebaker and J. Mitchell, 1993 (Van Nostrand Reinhold, New York) or "Discrete Cosine Transform: Algorithms, Advantages and Applications," by K. Rao and P. Yip, 1990 (Academic Press, San Diego).

The two-dimensional discrete cosine transform is a pair of mathematical equations that transforms one $N_1 \times N_2$ array of numbers to or from another $N_1 \times N_2$ array of numbers. The first array typically represents a square $N \times N$ array of spatially determined pixel values which form the digital image. The second array is an array of discrete cosine transform coefficients which represent the image in the frequency

5

10

15

20

transform for images use eight-by-eight arrays wherein N therefore has a value of eight. Assuming then that N has the value of eight when performing the transforms, where $f(i, j)$ are the values of the pixel array and $F(u, v)$ are the values of the discrete cosine transform coefficients, the equations of the discrete cosine transforms are as follows.

The formula for the 2D discrete cosine transform is given by:

$$F(u, v) = \frac{C_u C_v}{4} \sum_{x=0}^7 \sum_{y=0}^7 f(x, y) \cos\left(\frac{(2x+1)u\pi}{16}\right) \cos\left(\frac{(2y+1)v\pi}{16}\right)$$

where x, y = spatial coordinates in the spatial domain (0, 1, 2, ..., 7); u, v = coordinates in the transform domain (0, 1, 2, ..., 7); $C_u = \frac{1}{\sqrt{2}}$ for $u = 0$, otherwise 1; and $C_v = \frac{1}{\sqrt{2}}$ for $v = 0$, otherwise 1. The separable nature of the 2D DCT is exploited by performing a 1D DCT on the eight columns, and then a 1D DCT on the eight rows of the result. Several fast algorithms are available to calculate the 8-point 1D DCT.

As described above, a DCT compressor comprises mainly two parts. The first part transforms highly correlated image data into weakly correlated coefficients using a DCT transform and the second part performs quantization on coefficients to reduce the bit rate for transmission or storage. However, the computational burden in performing a DCT is demanding. For example, to process a one-dimensional DCT of length 8 pixels requires 13 multiplications and 29 additions in currently

known fast algorithms. As stated above, the image is divided into square blocks of size 8 by 8 pixels, 16 by 16 pixels or 32 by 32 pixels. Each block is often processed by the one-dimensional DCT in row-by-row fashion followed by column-by-column. On the other hand, different block sizes are selected for compression due to different types of input and different quality requirements on the decompressed data.

In the article, "A Fast DCT-SQ Scheme for Images," Trans. IEICE, Vol. E-71, No. 11, pp. 1095-1097, Nov. 1988, Y. Arai, T. Agui, and M. Nakajima proposed that many of the DCT multiplications can be formulated as scaling multipliers to the DCT coefficients. The DCT after the multipliers are factored out is called the scaled DCT.

The scaled DCT is still orthogonal but no longer normalized, whereas the scaling factors may be restored in a subsequent quantization process. Arai, et al. have demonstrated in their article that only 5 multiplications and 29 additions are required in processing an 8-point scaled DCT.

However, there is a need to further increase the speed of the encoder because more than half of the time in the JPEG encoder is spent in the Forward Discrete Cosine Transform (FDCT) code calculating the two-dimensional (2-D) 8x8 block of 8-bit or 12-bit samples. Currently, the 2-D FDCT is calculated by first calculating eight 1-D horizontal DCTs and then calculating another eight 1-D vertical DCTs using the currently fastest known 1-D DCT, which was first described by Arai, Agui, and Nakajima, as mentioned above. The current process takes 29 additions and 5 multiplication to calculate a scaled version of the 1-D FDCT. The scaling constants are the same for each vertical column and can finally be included in the quantization step. This prior solution saved eight multiplications per 1-D FDCT.

However, as stated above, it can be seen that there is a need to provide a faster DCT transform.

It can also be seen that there is a need to provide a method and apparatus for performing discrete cosine transforms with less addition and multiplication steps to

5 increase throughput of an encoder.

SUMMARY OF THE INVENTION

To overcome the limitations in the prior art described above, and to overcome other limitations that will become apparent upon reading and understanding the present specification, the present invention discloses a faster discrete cosine
5 transform that uses scaled terms.

The present invention solves the above-described problems by decreasing the number of multiplications by using scaling terms on the transform constants. Further, after scaling, the terms of the constants are replaced by a series of fewer linear shifts and additions since the constants may be approximated by sums of
10 powers-of-2 after only a few summations. Those skilled in the art will recognize that throughout this specification, the term 'matrix' is used in both its traditional mathematical sense and also to cover all hardware and software systems which when analyzed could be equivalently represented as a mathematical matrix.

A method in accordance with the principles of the present invention includes
15 arranging discrete cosine transform equations into at least one collection having at least two discrete cosine transform constants, scaling the discrete cosine transform equations in the at least one collection by dividing each of the discrete cosine transform constants in the collection by one of the discrete cosine transform constants from the at least one collection and representing each of the scaled discrete cosine
20 transform constants with estimated scaled discrete cosine transform constants approximated by sums of powers-of-2.

Other embodiments of a method in accordance with the principles of the invention may include alternative or optional additional aspects. One such aspect of

the present invention is that the method further includes separating an image into at least one block and transforming the block into transformed data by performing matrix multiplication on the discrete cosine transform equations based upon binary arithmetic using the estimated scaled discrete cosine transform constants and performing linear shifts and additions.

Another aspect of the present invention is that the scaling the discrete cosine transform equations in the at least one collection by dividing each of the discrete cosine transform constants in the at least one collection by one of the discrete cosine transform constants from the at least one collection saves multiplications.

Another aspect of the present invention is that the discrete cosine transform constant chosen for scaling the discrete cosine transform equations in the at least one collection is selected according to a predetermined cost function.

Another aspect of the present invention is that the cost function minimizes a number of add operations.

Another aspect of the present invention is that the cost function minimizes a worst case number of add operations.

Another aspect of the present invention is that the cost function minimizes an error per constant resulting from the approximations.

Another aspect of the present invention is that the transforming the block into transformed data further comprises generating at least one set of one dimensional discrete cosine transform equations.

Another aspect of the present invention is that the discrete cosine transform constants are obtained by splitting the discrete cosine transform constants into even and odd terms by obtaining sums and differences of input samples.

Another aspect of the present invention is that the block is an $N_1 \times N_2$ block.

5 Another aspect of the present invention is that $N_1 = N_2 = 8$.

In another embodiment of the present invention, a data compression system is provided. The data compression system includes a discrete cosine transformer for applying a discrete cosine transform to decorrelate data into discrete cosine transform equations, the discrete cosine transform equations having been formed by
10 arranging the discrete cosine transform equations into at least one collection having at least two discrete cosine transform constants, scaling the discrete cosine transform equations in the at least one collection by dividing each of the discrete cosine transform constant in the collection by one of the discrete cosine transform constants from the at least one collection and representing each of the scaled
15 discrete cosine transform constants with estimated scaled discrete cosine transform constants approximated by sums of powers-of-2 and a quantizer for quantizing the transformed data into quantized data to reduce the number of bits needed to represent the transform coefficients.

In another embodiment of the present invention, a printer is provided. The
20 printer includes a memory for storing data, a processor for processing the data to provide a compressed print stream output and a printhead driving circuit for controlling a printhead to generate a printout of the data, wherein the processor applies a discrete cosine transform to decorrelate data into transform coefficients

using discrete cosine equations, the discrete cosine transform equations having been formed by arranging the discrete cosine transform equations into at least one collection having at least two discrete cosine transform constants, scaling the discrete cosine transform equations in the at least one collection by dividing each of the discrete cosine transform constant in the collection by one of the discrete cosine transform constants from the at least one collection and representing each of the scaled discrete cosine transform constants with estimated scaled discrete cosine transform constants approximated by sums of powers-of-2.

In another embodiment of the present invention, an article of manufacture is provided. The article of manufacture includes a program storage medium readable by a computer, the medium tangibly embodying one or more programs of instructions executable by the computer to perform a method for performing faster discrete cosine transform, the method includes arranging discrete cosine transform equations into at least one collection having at least two discrete cosine transform constants, scaling the discrete cosine transform equations in the at least one collection by dividing each of the discrete cosine transform constant in the collection by one of the discrete cosine transform constants from the at least one collection and representing each of the scaled discrete cosine transform constants with estimated scaled discrete cosine transform constants approximated by sums of powers-of-2.

In another embodiment of the present invention, a data analysis system is provided. The data analysis system includes a memory for storing the discrete cosine transform equations having been formed by arranging the discrete cosine

transform equations into at least one collection having at least two discrete cosine transform constants, scaling the discrete cosine transform equations in the at least one collection by dividing each of the discrete cosine transform constant in the collection by one of the discrete cosine transform constants from the at least one collection and representing each of the scaled discrete cosine transform constants with estimated scaled discrete cosine transform constants approximated by sums of powers-of-2 and a transformer for applying the transform equations to perform a discrete cosine transform to decorrelate data into discrete cosine transform coefficients.

10 These and various other advantages and features of novelty which characterize the invention are pointed out with particularity in the claims annexed hereto and form a part hereof. However, for a better understanding of the invention, its advantages, and the objects obtained by its use, reference should be made to the drawings which form a further part hereof, and to accompanying descriptive matter, in which there are
15 illustrated and described specific examples of an apparatus in accordance with the invention.

BRIEF DESCRIPTION OF THE DRAWINGS

Referring now to the drawings in which like reference numbers represent corresponding parts throughout:

Fig. 1 illustrates a block diagram of a JPEG encoder;

5 Fig. 2 illustrates a table that shows the terms C2 and C6;

Fig. 3 is a table that shows the actual values if either C2 or C6 is used to scale the coefficient matrix and the estimates are a ratio of integers;

Fig. 4 shows all possible combinations of dividing the coefficient matrix by C1, C3, C5, and C7;

10 Fig. 5 shows the actual values when C5 is used to scale the coefficient matrix;

Fig. 6 shows the actual values when C7 is used to scale the coefficient matrix;

15 Fig. 7 shows the number of binary adds to generate the unscaled cosine terms within one percent error;

Fig. 8 illustrates a flow chart of the method to design the fast DCT using scaled terms according to the present invention;

Fig. 9 illustrates a printer according to the present invention;

Fig. 10 illustrates a data analyzing system according to the present invention;

20 and

Fig. 11 illustrates another data analyzing system according to the present invention.

DETAILED DESCRIPTION OF THE INVENTION

In the following description of the exemplary embodiment, reference is made to the accompanying drawings which form a part hereof, and in which is shown by way of illustration the specific embodiment in which the invention may be practiced.

5 It is to be understood that other embodiments may be utilized as structural changes may be made without departing from the scope of the present invention.

The present invention provides a faster discrete cosine transform that uses scaled terms. The present invention decreases the number of multiplications by using scaled terms for the transform constants. Further, after scaling, the ratios of
10 the constants are replaced by a series of linear shifts and additions since the constants may be approximated by sums of powers-of-2 after only a few summations.

Fig. 1 illustrates a block diagram of a JPEG encoder 100. In Fig. 1, digital image data 110 is divided up into 8 by 8 pixel blocks 112. Then, the discrete cosine transform (DCT) of each block is calculated 120. The discrete cosine transform
15 (DCT) helps separate the data into parts (or spectral sub-bands) of differing importance (e.g., with respect to an image's visual quality). The DCT is similar to the discrete Fourier transform: it transforms a signal from the spatial domain to the frequency domain.

20 A quantizer 130 rounds off the DCT coefficients according to the quantization matrix. This step produces the "lossy" nature of JPEG, but allows for large compression ratios. There is a tradeoff between image quality and degree of quantization. A large quantization step size can produce unacceptably large image

distortion. This effect is similar to quantizing Fourier series coefficients too coarsely; large distortion would result. Unfortunately, finer quantization leads to lower compression ratios. The question is how to quantize the DCT coefficients most efficiently. Because of human eyesight's natural high frequency roll-off, these frequencies play a less important role than low frequencies. This lets JPEG use a much higher step size for the high frequency coefficients, with little noticeable image deterioration. The quantizer coefficient output may then be encoded by an optional entropy encoder 140 to produce a compressed data stream 150 to an output file. Note that for JPEG the entropy encoder is not optional, but other similar data compression systems could be designed without the CPU cycles required by the entropy encoder.

However, as described below, there is a need to provide a method and apparatus for performing discrete cosine transforms with less multiplication steps to increase throughput of the encoder 100. As will be described, the method according to the present invention saves multiplications in the brute force equations by scaling the coefficient matrix. Each separable subgroup is scaled independently of the other collections. Within each collection, the remaining multiplications are replaced by simple shifts and adds. The scaling terms are chosen according to various cost functions. The preferred embodiment uses cost functions that minimize the number of adds and that minimize the worst case number of adds. However, those skilled in the art will recognize that alternate cost functions could choose how much error is allowed per coefficient. Moreover, those skilled in the art will recognize that the

inverse DCT (IDCT) can be implemented using the same method so that the same number of operations are used.

Let the $C_n = \cosine(n * \pi/16)$. Let $f(x)$ for $x = 0$ to 7 be the input samples. Let $F(u)$ for $u=0$ to 7 be proportional to the 1-D transform coefficients. Let $S(u)$ for $u=0$ to 7 be equal to the 1-D transform coefficients. Let s_{xy} be $f(x) + f(y)$ and d_{xy} be $f(x) - f(y)$ for $x, y = 0$ to 7 . Note the calculations for $F(u)$ below ignore some constants that end up in the quantization terms. The first step in the 1-D DCT is to split the coefficients into the even and odd terms by taking sums and differences of the eight input samples.

10

$$s_{07} = f(0) + f(7)$$

$$s_{16} = f(1) + f(6)$$

$$s_{25} = f(2) + f(5)$$

$$s_{34} = f(3) + f(4)$$

15

$$d_{07} = f(0) - f(7)$$

$$d_{16} = f(1) - f(6)$$

$$d_{25} = f(2) - f(5)$$

$$d_{34} = f(3) - f(4)$$

20

Then, the sums and differences of the sums are calculated:

$$s_{0734} = s_{07} + s_{34}$$

$$s_{1625} = s_{16} + s_{25}$$

$$d_{0734} = s_{07} - s_{34}$$

$$d_{1625} = s_{16} - s_{25}$$

25

Finally we take sums and differences of the sums again to calculate $F(0)$ and $F(4)$.

$$F(0) = s_{0734} + s_{1625}$$

$$F(4) = s_{0734} - s_{1625}$$

30

$$2 * S(0) = C_4(s_{0734} + s_{1625})$$

$$2 * S(4) = C_4(s_{0734} - s_{1625})$$

Those skilled in the art will recognize that the equations above have been derived similar to the Arai, Agui, and Nakajima algorithm. The F(2) and F(6) can be calculated from these terms as:

$$\begin{aligned}
 5 \quad & F(2) = C2 * d0734 + C6 * d1625 \\
 & F(6) = C6 * d0734 - C2 * d1625 \\
 & 2 * S(2) = C2 * d0734 + C6 * d1625 \\
 & 2 * S(6) = C6 * d0734 - C2 * d1625
 \end{aligned}$$

10 Fig. 2 illustrates a table 200 that shows the C2 and C6 terms divided by C2 or C6. C2 is equal to 0.923880, C2/C2 is equal to 1.000000, C2/C6 is equal to $1 + \sqrt{2}$, C6 is equal to 0.382683, C6/C2 is equal to $\frac{1}{1 + \sqrt{2}}$, and C6/C6 is equal to 1.000000.

Next, the utility of scaling the constants in the matrix by C6 or C2 is investigated.

Fig. 3 is a table 300 that shows the actual values 310 if either C2 or C6 is used to scale the constants in the matrix and the estimates 312 are a ratio of integers. Fig. 3 shows the error 314 due to each of the approximations. C2 was chosen because the greatest error occurs in the harmonic with the least power (or the smallest contribution to the overall output).

20 Since F(2) is much more likely to occur, it is now exactly calculated. So the equations become:

$$\begin{aligned}
 & F(2) = 12 * d0734 + 5 * d1625 = d1625 + (d0734 + d1625 + (d0734 << 1)) << 2 \\
 & F(6) = 5 * d0734 - 12 * d1625 = d0734 + (d0734 - d1625 - (d1625 << 1)) << 2 \\
 & (24/C2) * S(2) = d1625 + (d0734 + d1625 + (d0734 << 1)) << 2 \\
 25 \quad & (24/C2) * S(6) = d0734 + (d0734 - d1625 - (d1625 << 1)) << 2
 \end{aligned}$$

Equation 1 shows the remaining equations for the odd constants expressed in

matrix notation.

$$\begin{bmatrix} F(1) \\ F(3) \\ F(5) \\ F(7) \end{bmatrix} = \begin{bmatrix} +C1 & +C3 & +C5 & +C7 \\ +C3 & -C7 & -C1 & -C5 \\ +C5 & -C1 & +C7 & +C3 \\ +C7 & -C5 & +C3 & -C1 \end{bmatrix} \begin{bmatrix} d07 \\ d16 \\ d25 \\ d34 \end{bmatrix} \quad \text{Equation 1}$$

5 In Equation 1, F(1), F(3), F(5) and F(7) are a vector related to unquantized 1-D DCT transform coefficients. Note: for $u = (1, 3, 5, 7)$, $2 * S(u) = F(u)$. If the solutions for these four coefficients are found using a "brute force" approach, then 12 additional additions and 16 multiplications would be necessary. However, if the matrix of coefficients is divided by one of the constants in the matrix, then the
10 number of multiplications necessary is reduced by four. This linear system is shown in Equation 2 below.

$$\frac{1}{C5} * \begin{bmatrix} F(1) \\ F(3) \\ F(5) \\ F(7) \end{bmatrix} = \begin{bmatrix} +C1/C5 & +C3/C5 & +1 & +C7/C5 \\ +C3/C5 & -C7/C5 & -C1/C5 & -1 \\ +1 & -C1/C5 & +C7/C5 & +C3/C5 \\ +C7/C5 & -1 & +C3/C5 & -C1/C5 \end{bmatrix} \begin{bmatrix} d07 \\ d16 \\ d25 \\ d34 \end{bmatrix} \quad \text{Equation 2}$$

15 Equation 2 shows that the linear set now only involves 12 additions, 12 multiplications and one division. However, the constant C5 in the division can be included in the quantization numbers once per image. Meanwhile, the benefits of only 12 multiplications instead of 16 are reaped every 8 by 8 block of samples. Furthermore, when the constants of the matrix were scaled by one of the terms, all
20 of the terms became approximations of sums of powers-of-2 within a few successive approximations. This is very beneficial because a multiplication which normally is

considered to necessitate more than 5 CPU cycles now becomes a shift and add with a maximum of 3 shifts and 2 additions. Fig. 4 shows all possible combinations of dividing the constants in the matrix by C1, C3, C5, and C7.

Below, the utility of scaling the constants in the matrix by C5 or by C7 is investigated. Fig. 5 is a table 500 showing the actual values 510 if C5 is used to scale the constants of the matrix. The estimates 512 are in sums of powers-of-2, and the error 514 due to the approximation is also stated. C5 was chosen because of the continuity in the terms. It was also chosen because the greatest error occurs in the harmonic with the least power (or the smallest contribution to the overall output). The constant C7 is also a good candidate because the calculations fit inside of 16 bits. This optimization is shown in table 600 of Fig. 6.

Those skilled in the art will recognize the above scaling is not meant to be an optimal solution for minimizing error. Nevertheless, an exhaustive search may be performed to find the optimal number to scale the matrix. Also, the scaling factor need not be a function of $\cos(n * \pi/16)$. As shown for the C2 and C6 cases, the best answer was a ratio of integers. For comparison purposes, Fig. 7 is a table 700 showing the number of binary adds and used to calculate the unscaled cosine terms within one percent error.

In summary, the scaled terms approach fundamentally decreases the number of multiplications because the scaling on the quantization values is done once per image. However, the benefits are reaped every 8x8 block, or more generally, $N_1 \times N_2$ block. Second, after scaling, the ratios of coefficients are very nearly sums of powers-of-2 after only a few summations. Thus, multiplications can be replaced by a

series of linear shifts and additions. In some cases, these terms may be further converted into integer terms. This will provide superior hardware and very fast software on machines where multiplications are costly (in terms of CPU cycles).

Fig. 8 illustrates a flow chart 800 of the method used to design the fast DCT method using scaled terms according to the present invention. In Fig. 8, the one dimensional transform coefficients equations for the DCTs are obtained 810. To obtain the coefficients, the coefficients are split into even and odd terms by obtaining sums and differences with the eight input samples, which are then used to calculate the one dimensional transform coefficients.

After the one dimensional coefficients are obtained, the constants in the 1-D transform matrix are scaled by dividing by one of the constants 820. This reduces the number of multiplications, for an 8x8 block, by four. After scaling, the ratios of constants are very nearly sums of powers-of-2 after only a few summations. Thus, multiplications can be replaced by a series of linear shifts and additions 830.

Fig. 9 illustrates a block diagram 900 of a printer 920 according to the present invention. In Fig. 9, the printer 920 receives data 912 from a host processor 910. The data 912 is provided into memory 930 where the data 912 may be arranged into blocks. The blocks are then processed by a processor 940, such as a raster image processor. The processor 940 provides a compressed print stream representing the data to a printhead driving circuit 950. The printhead driving circuit 950 then controls the printhead 960 to generate a printout 970 of the data.

The algorithm designed with reference to Fig. 8 may be tangibly embodied in a computer-readable medium or carrier 990, e.g. one or more of the fixed and/or

removable data storage devices illustrated in Fig. 9, or other data storage or data communications devices. The computer program may be loaded into the memory 942 to configure the processor 940 of Fig. 9, for execution. The computer program comprise instructions which, when read and executed by the processor 940 of Fig. 9, causes the processor 940 to perform the steps necessary to execute the steps or elements of the present invention.

Fig. 10 illustrates a data analyzing system 1000 according to the present invention. In Fig. 10, a discrete cosine transform 1010 receives a block of data 1012 to be analyzed. The discrete cosine transform 1010 uses discrete cosine transform equations 1020 to generate transformed data 1024. Prior to execution, the discrete cosine transform equations 1020 are split into at least one sub-transform having at least two transform constants. The at least two transform constants for each collection are independently scaled with a scaling term to maintain a substantially uniform ratio between the at least two transform constants within the at least one collection, wherein the scaling term is chosen according to a predetermined cost function. The discrete cosine transformed data 1024 may then be quantized by an optional quantizer 1030.

Fig. 11 illustrates another data analyzing system 1100 according to the present invention. In Fig. 11, a discrete cosine transform 1110 receives a block of data 1112 to be analyzed. The discrete cosine transform 1110 uses discrete cosine transform equations 1120 to generate transformed data 1124. Prior to execution, the discrete cosine transform equations 1120 are split into at least one sub-transform having at least two transform constants. The at least two transform

constants for each collection are independently scaled with a scaling term to maintain a substantially uniform ratio between the at least two transform constants within the at least one collection, wherein the scaling term may be chosen according to a predetermined cost function. The discrete cosine transformed data 1124 may
5 then be compared to scaled comparison values in an optional comparator 1130.

The foregoing description of the exemplary embodiment of the invention has been presented for the purposes of illustration and description. It is not intended to be exhaustive or to limit the invention to the precise form disclosed. Many modifications and variations are possible in light of the above teaching. It is
10 intended that the scope of the invention be limited not with this detailed description, but rather by the claims appended hereto.

11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000